

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-337935

(43)Date of publication of application : 07.12.2001

(51)Int.Cl.

G06F 15/16

G06F 9/44

(21)Application number : 2000-153678

(71)Applicant : NEC CORP

(22)Date of filing : 24.05.2000

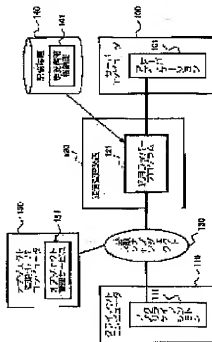
(72)Inventor : NISHIDA TAKESHI

(54) APPLICATION WRAPPING METHOD IN DISPERSED OBJECT ENVIRONMENT, ITS COMMUNICATION MANAGEMENT DEVICE, AND RECORDING MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To update wrapper program operation without any alteration of a wrapper program itself even when definition information necessary for wrapping is changed in wrapping over a dispersed object environment of application on a non-dispersed object environment.

SOLUTION: When a general-purpose wrapper program 121 refers to a definition information storage part 141 storing definition information necessary for wrapping, operation of the general-purpose wrapper program 121 is decided. When information necessary for the wrapping such as server application 101 to be wrapped and an interface for a client application 111 is changed, the definition information in the definition information storage part 141 is changed without changing any general-purpose wrapper program 121 itself, so that operation of the general-purpose wrapper program 121 can be changed.



LEGAL STATUS

[Date of request for examination] 13.04.2001

[Date of sending the examiner's decision of rejection] 12.11.2002

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

【特許請求の範囲】

【請求項1】 非分散オブジェクト環境上に存在する少なくとも1個のアプリケーションを分散オブジェクト環境上のオブジェクトとして利用できるように、前記それぞれのアプリケーション間で処理の受け渡しを行う分散オブジェクト環境におけるアプリケーションのラッピング方法であって、前記非分散オブジェクト環境上に存在するアプリケーションを前記分散オブジェクト環境上のオブジェクトとしてラッピングするために必要な情報を参照することによって前記非分散オブジェクト環境上に存在するアプリケーションのラッピングの方法を決定し、その方法に従って動作を決定することと特徴とする分散オブジェクト環境におけるアプリケーションのラッピング方法。

【請求項2】 前記ラッピングは、前記分散オブジェクト環境上の利用可能オブジェクトを管理しているオブジェクト管理サービスに対して前記非分散オブジェクト環境上に存在するアプリケーションの登録もしくは削除を行い、前記分散オブジェクト環境上に存在するアプリケーションから送られてきた電文の形式を前記非分散オブジェクト環境上に存在するアプリケーションへ送る電文形式に変換し、前記非分散オブジェクト環境上に存在するアプリケーションから送られてきた電文の形式を前記分散オブジェクト環境上に存在するアプリケーションへ送る電文形式に変換することと特徴とする請求項1に記載の分散オブジェクト環境におけるアプリケーションのラッピング方法。

【請求項3】 更に、前記ラッピングに必要な情報から前記分散オブジェクト環境上に存在するアプリケーションの作成を支援するインターフェース定義情報を作成することと特徴とする請求項1または請求項2に記載の分散オブジェクト環境におけるアプリケーションのラッピング方法。

【請求項4】 非分散オブジェクト環境上に存在する少なくとも1個のアプリケーションを分散オブジェクト環境上のオブジェクトとして利用できるように、前記それぞれのアプリケーション間で処理の受け渡しを行う通信管理装置であって、前記非分散オブジェクト環境上に存在するアプリケーションを分散オブジェクト環境上に存在するオブジェクトとしてラッピングするために必要な情報を参照することによって前記非分散オブジェクト環境上に存在するアプリケーションのラッピング方法を決定し、その方法に従って動作を決定するラッピング手段を備えたことを特徴とする通信管理装置。

【請求項5】 前記ラッピング手段は、前記分散オブジェクト環境上に存在する利用可能オブジェクトを管理しているオブジェクト管理サービスに対して前記非分散オブジェクト環境上に存在するアプリケーションの登録もしくは削除をする手段と、前記分散オブジェクト環境上に存在するアプリケーションから送られてきた電文の形

式を前記非分散オブジェクト環境上に存在するアプリケーションへ送る電文形式に変換する手段と、前記非分散オブジェクト環境上に存在するアプリケーションから送られてきた電文の形式を前記分散オブジェクト環境上に存在するアプリケーションへ送る電文形式に変換する手段とを含むことを特徴とする請求項4に記載の通信管理装置。

【請求項6】 前記ラッピング手段は、前記ラッピングに必要な情報から分散オブジェクト環境上に存在するアプリケーションの作成を支援するインターフェース定義情報を作成する手段を含むことを特徴とする請求項4または請求項5に記載の通信管理装置。

【請求項7】 非分散オブジェクト環境上に存在する少なくとも1個のアプリケーションを分散オブジェクト環境上のオブジェクトとして利用できるように、前記それぞれのアプリケーション間で処理の受け渡しを行う、分散オブジェクト環境におけるアプリケーションのラッピング処理を行うプログラムであって、前記非分散オブジェクト環境上に存在するアプリケーションを前記分散オブジェクト環境上のオブジェクトとしてラッピングするために必要な情報を参照するステップと、前記参照ステップによって前記非分散オブジェクト環境上に存在するアプリケーションのラッピング方法を決定し、それに従って動作を決定するステップとがプログラムされ記録されたコンピュータ読み取り可能な記録媒体。

【請求項8】 前記ラッピング方法を決定し、それに従って動作を決定するステップは、前記分散オブジェクト環境上の利用可能オブジェクトを管理しているオブジェクト管理サービスに対して前記非分散オブジェクト環境上に存在するアプリケーションの登録もしくは削除を行うステップと、前記分散オブジェクト環境上のアプリケーションから送られてきた電文の形式を前記非分散オブジェクト環境上に存在するアプリケーションへ送る電文形式に変換するステップと、前記非分散オブジェクト環境上に存在するアプリケーションから送られてきた電文の形式を前記分散オブジェクト環境上のアプリケーションへ送る電文形式に変換するステップとを含むことを特徴とするコンピュータ読み取り可能な請求項7に記載の記録媒体。

【請求項9】 前記ラッピング方法を決定し、それに従って動作を決定するステップは、更に、ラッピングに必要な情報から分散オブジェクト環境上のアプリケーションの作成を支援するためのインターフェース定義情報を作成するステップを含むことを特徴とするコンピュータ読み取り可能な請求項7または請求項8に記載の記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、分散オブジェクト環境におけるアプリケーションのラッピング方法ならびにその通信管理装置および同方法がプログラムされ記録

された記録媒体に關し、特に、非分散オブジェクト環境上に存在するアプリケーションと分散オブジェクト環境上のアプリケーションとの間で、一方が他方に処理を要求し、処理を要求されたアプリケーションが処理を要求したアプリケーションに処理結果を返すことのできる、分散オブジェクト環境におけるアプリケーションのラッピング方法ならびにその通信管理装置および同方法がプログラムされ記録された記録媒体に關する。

【0002】

【従来の技術】OMG (Object Management Group) 10 が提唱しているCORBA (Common Object Request Broker Architecture) に代表される分散オブジェクト環境において、例えば汎用コンピュータ上のランザクションシステムのような特定の非分散オブジェクト環境のアプリケーションを分散オブジェクト環境上で利用する方法として、非分散オブジェクト環境のアプリケーションへの入出力を分散オブジェクト環境上のインターフェースに変換する方法がある。上記のようにして、非分散オブジェクト環境上のアプリケーションを分散オブジェクト環境上のオブジェクトにみせかけることをラッピングといい、また、これを実現する機能部をラッパ

ーという。

【0003】ラッピングシステムの一例を図8に示す。図に示すように、ラッピングシステムは、サーバコンピュータ100と、クライアントコンピュータ110と、通信管理装置120と、分散オブジェクトネットワーク130と、記憶装置140と、オブジェクト管理サーバコンピュータ150とから構成されている。サーバコンピュータ100は、サーバアプリケーション101を備えている。サーバアプリケーション101は、サーバコンピュータ100の外部への入出力インターフェースを持ち、外部からの要求に応じて特定の処理を実行し、その実行結果を返すプログラムである。クライアントコンピュータ110は、クライアントアプリケーション111を備えている。クライアントアプリケーション111は、クライアントコンピュータ110の外部への入出力インターフェースを持ち、サーバアプリケーション101に処理を要求し、その結果を反映するような、ある特定の処理を行うプログラムである。

【0004】通信管理装置120は、ラッパープログラム122を備えている。ラッパープログラム122は、ラッパーソースコード生成手段123およびコンパイラ125によって生成され、クライアントアプリケーション111のインターフェースとサーバアプリケーション101のインターフェースとの間の変換と、通信管理を行うプログラムである。記憶装置140は、定義情報格納部141を備えている。定義情報格納部141は、サーバアプリケーション101のインターフェース情報などといった、非分散オブジェクト環境上のアプリケーションに対するラッピングに必要な定義情報を格納してい

る機能部である。

【0005】オブジェクト管理サーバコンピュータ150は、オブジェクト管理サービス151を備えている。オブジェクト管理サービス151は、分散オブジェクト環境上のオブジェクトを一意的に識別するための情報を管理するプログラムで、クライアントプログラムは、オブジェクト管理サービス151に問い合わせることによって、オブジェクトを一意的に識別するための情報を取得することができ、この情報をもちいることで、目的のオブジェクトに処理を要求することができる。

【0006】このような構成を有する従来のラッピングシステムは次のように動作する。まず、ラッパーソースコード生成手段123は、定義情報格納部141に格納されている定義情報を参照して、ラッパーソースコード124を生成する。そして、ラッパーソースコード124をコンパイラ125でコンパイル、リンクすること

で、ラッパープログラム122を生成する。、

【0007】生成されたラッパープログラム122は、サーバアプリケーション101用のオブジェクトをオブジェクト管理サービス151に登録する。クライアントアプリケーション111は、オブジェクト管理サービス151にあるオブジェクト情報を参照することで、ラッパープログラム122と通信を行う。クライアントアプリケーション111から入力データを受け取ったラッパープログラム122は、その入力データをサーバアプリケーション101の入力形式に変換して、サーバアプリケーション101に渡す。サーバアプリケーション101がラッパープログラム122に出力データを返すと、ラッパープログラム122は、その出力データをクライアントアプリケーション111が要求している形式に変換して返す。

【0008】以上の処理を行うことで、分散オブジェクト環境上にあるクライアントアプリケーション111から非分散オブジェクト環境下のサーバアプリケーション101を利用することが可能になる。

【0009】

【発明が解決しようとする課題】上述した従来技術によれば、ラッピング対象となる非分散オブジェクト環境上のアプリケーションのインターフェースや、非分散オブジェクト環境上のアプリケーションとの処理の受け渡しを行う分散オブジェクト環境下のアプリケーション、非分散オブジェクト環境や分散オブジェクト環境上の通信に關する定義など、ラッパーが必要とする定義情報が1つでも変化するとき、ラッパープログラムそのものも再度生成しなおさなければならない。すなわち、従来のラッピング方法によれば、ラッピングに必要な各定義情報を基にして、それに対応したラッパープログラムのソースコードを生成し、それをコンパイル、リンクするという方式をとっていたため、ラッパーが必要とする定義情報が1つでも変化するとき、ラッパープログラムそ

のものも再度生成し直す必要がある。したがって、プログラマに対する負担が大きく、効率が悪かった。

【0010】本発明は上記事情に鑑みてなされたものであり、非分散オブジェクト環境上に存在するアプリケーションを前記分散オブジェクト環境上のオブジェクトとしてラッピングするために必要な情報を参照することによって前記非分散オブジェクト環境上に存在するアプリケーションのラッピングの方法を決定し、その方法に従って動作を決定することにより、ラッピングに必要な定義情報に変更が生じても、ラッパープログラムのオブジェクトコードを再度生成することなしに、ラッパープログラムの動作を更新することができる、分散オブジェクト環境におけるアプリケーションのラッピング方法ならびにその通信管理装置および同方法がプログラムされ記録された記録媒体を提供することを目的とする。

【0011】

【課題を解決するための手段】上述した課題を解決するために請求項1に記載の発明は、非分散オブジェクト環境上に存在する少なくとも1個のアプリケーションを分散オブジェクト環境上のオブジェクトとして利用できるように、それぞれのアプリケーション間で処理の受け渡しを行う分散オブジェクト環境におけるアプリケーションのラッピング方法であって、非分散オブジェクト環境上に存在するアプリケーションを分散オブジェクト環境上のオブジェクトとしてラッピングするために必要な情報を参照することによって非分散オブジェクト環境上に存在するアプリケーションのラッピングの方法を決定し、その方法に従って動作を決定することとした。このことにより、ラッパープログラムがインターフェースの定義情報を参照して動作するため、その定義情報を変更するだけで、インターフェースの変更がラッパープログラムに反映され、したがって、非分散もしくは分散オブジェクト環境内のアプリケーションのインターフェースに変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、ラッパープログラムに対してインターフェースの変更を反映できる。

【0012】請求項2に記載の発明は、請求項1に記載の分散オブジェクト環境におけるアプリケーションのラッピング方法において、ラッピングは、分散オブジェクト環境上の利用可能オブジェクトを管理しているオブジェクト管理サービスに対して非分散オブジェクト環境上に存在するアプリケーションの登録もしくは削除を行い、分散オブジェクト環境上に存在するアプリケーションから送られてきた電文の形式を前記非分散オブジェクト環境上に存在するアプリケーションへ送る電文形式に変換し、非分散オブジェクト環境上に存在するアプリケーションから送られてきた電文の形式を前記分散オブジェクト環境上に存在するアプリケーションへ送る電文形式に変換することとした。このことにより、ラッパープログラムが分散オブジェクト環境内および非分散オブ

ジェクト環境内の通信に関する定義情報を参照して動作するため、その定義情報を変更するだけで、通信定義の変更がラッパープログラムに反映され、分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義に変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、ラッパープログラムに対して通信定義の変更を反映できる。

【0013】請求項3に記載の発明は、請求項1または請求項2に記載の分散オブジェクト環境におけるアプリケーションのラッピング方法において、更に、ラッピングに必要な情報から分散オブジェクト環境上に存在するアプリケーションの作成を支援するインターフェース定義情報を作成することとした。このことにより、あらかじめ格納されているインターフェース情報から、分散オブジェクト環境上でのインターフェース定義である、例えばDLを自動的に作成することによって、非分散オブジェクト環境上のアプリケーションを利用する分散オブジェクト環境内のクライアントアプリケーションを作成する手間を軽減することができることもできる。

【0014】請求項4に記載の発明は、非分散オブジェクト環境上に存在する少なくとも1個のアプリケーションを分散オブジェクト環境上のオブジェクトとして利用できるように、それぞれのアプリケーション間で処理の受け渡しを行う通信管理装置であって、非分散オブジェクト環境上に存在するアプリケーションを分散オブジェクト環境上に存在するオブジェクトとしてラッピングするために必要な情報を参照することによって非分散オブジェクト環境上に存在するアプリケーションのラッピング方法を決定し、その方法に従って動作を決定するラッピング手段を備えることとした。上記構成により、ラッパープログラムがインターフェースの定義情報を参照して動作するため、その定義情報を変更するだけで、インターフェースの変更がラッパープログラムに反映され、したがって、非分散もしくは分散オブジェクト環境内のアプリケーションのインターフェースに変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、ラッパープログラムに対してインターフェースの変更を反映できる。このため、非分散もしくは分散オブジェクト環境内のアプリケーションのインターフェースに変更が生じた際にプログラマの負担を大幅に軽減したような新管理装置を提供でき、効率のよいラッピングシステムを構築できる。

【0015】請求項5に記載の発明は、請求項4に記載の通信管理装置において、ラッピング手段は、分散オブジェクト環境上に存在する利用可能オブジェクトを管理しているオブジェクト管理サービスに対して前記非分散オブジェクト環境上に存在するアプリケーションの登録もしくは削除をする手段と、分散オブジェクト環境上に存在するアプリケーションから送られてきた電文の形式を前記非分散オブジェクト環境上に存在するアプリケー

7
 ションへ送る電文形式に変換する手段と、非分散オブジェクト環境上にあるアプリケーションから送られてきた電文の形式を前記分散オブジェクト環境上に存在するアプリケーションへ送る電文形式に変換する手段とを含むこととした。上記構成により、ラッパープログラムが分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義情報を参照して動作するため、その定義情報を変更するだけで、通信定義の変更がラッパープログラムに反映されことから、分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義に変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、ラッパープログラムに対して通信定義の変更を反映できる。このため、分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義に変更が生じた際にプログラムの負担を大幅に軽減した通信管理装置を提供でき、効率のよいラッピングシステムを構築できる。

【0016】請求項6に記載の発明は、請求項4または請求項5に記載の通信管理装置において、ラッピング手段は、ラッピングに必要な情報から分散オブジェクト環境上に存在するアプリケーションの作成を支援するインターフェース定義情報を作成する手段を含むこととした。上記構成により、あらかじめ格納されているインターフェース情報から、分散オブジェクト環境上でのインターフェース定義である、例えばIDLを自動的に作成することによって、非分散オブジェクト環境上のアプリケーションを利用する分散オブジェクト環境内のクライアントアプリケーションを作成する手間を軽減した通信管理装置を提供でき、また、効率のよいラッピングシステムを構築できる。

【0017】請求項7に記載の発明は、非分散オブジェクト環境上に存在する少なくとも1個のアプリケーションを分散オブジェクト環境上のオブジェクトとして利用できるように、それぞれのアプリケーション間で処理の受け渡しを行う、分散オブジェクト環境におけるアプリケーションのラッピング処理を行うプログラムであって、非分散オブジェクト環境上に存在するアプリケーションを前記分散オブジェクト環境上のオブジェクトとしてラッピングするために必要な情報を参照するステップと、参照ステップによって前記非分散オブジェクト環境上に存在するアプリケーションのラッピング方法を決定し、それによって動作を決定するステップとがプログラムされ記録されることとした。上記したプログラムをコンピュータが読み出し実行することにより、ラッパープログラムがインターフェースの定義情報を参照して動作するため、その定義情報を変更するだけで、インターフェースの変更がラッパープログラムに反映され、したがって、非分散もしくは分散オブジェクト環境内のアプリケーションのインターフェースに変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、

ラッパープログラムに対してインターフェースの変更を反映できる。このため、分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義に変更が生じた際にプログラムの負担を大幅に軽減することができる。

【0018】請求項8に記載の発明は、請求項7に記載の記録媒体において、ラッピング方法を決定し、それによって動作を決定するステップは、分散オブジェクト環境上の利用可能なオブジェクトを管理しているオブジェクト管理サービスに対して非分散オブジェクト環境上にあるアプリケーションの登録もしくは削除を行うステップと、分散オブジェクト環境上のアプリケーションから送られてきた電文の形式を非分散オブジェクト環境上にあるアプリケーションへ送る電文形式に変換するステップと、非分散オブジェクト環境上にあるアプリケーションから送られてきた電文の形式を分散オブジェクト環境上のアプリケーションへ送る電文形式に変換するステップとを含むこととした。上記プログラムをコンピュータが読み出し実行することにより、ラッパープログラムが分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義情報を参照して動作するため、その定義情報を変更するだけで、通信定義の変更がラッパープログラムに反映されことから、分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義に変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、ラッパープログラムに対して通信定義の変更を反映できる。このため、分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義に変更が生じた際にプログラムの負担を大幅に軽減することができる。

【0019】請求項9に記載の発明は、請求項7または請求項8に記載の記録媒体において、ラッピング方法を決定し、それによって動作を決定するステップは、更に、ラッピングに必要な情報から分散オブジェクト環境上のアプリケーションの作成を支援するためのインターフェース定義情報を作成するステップを含むこととした。上記プログラムをコンピュータが読み出し実行することにより、あらかじめ格納されているインターフェース情報から、分散オブジェクト環境上でのインターフェース定義である、例えばIDLを自動的に作成することによって、非分散オブジェクト環境上のアプリケーションを利用する分散オブジェクト環境内のクライアントアプリケーションを作成する手間を軽減することができる。

【0020】

【発明の実施の形態】以下、図面を使用して本発明の実施形態について詳細に説明する。図1は本発明が採用されるラッピングシステムの一実施形態を示すブロック図である。図1において、ラッピングシステムは、プログラム制御により動作するサーバコンピュータ100と、クライアントコンピュータ110と、通信管理装置120

と、分散オブジェクトネットワーク130、記憶装置140、オブジェクト管理サーバコンピュータ150とから構成されている。サーバコンピュータ100は、サーバアプリケーション101を含む、クライアントコンピュータ110は、クライアントアプリケーション111を含む。また、通信管理装置120は、汎用ラッパープログラム121を含む、記憶装置140は、定義情報格納部141を含む。更に、オブジェクト管理サーバコンピュータ150は、オブジェクト管理サービス151を含む。

【0021】サーバアプリケーション101は、サーバコンピュータ100の外部との入力インターフェースもしくは入力インターフェースと出力インターフェースの両方を持ち、外部からの要求に応じてある特定の処理を実行する。クライアントアプリケーション111は、クライアントコンピュータ110上で動作し、サーバアプリケーション101に対して分散オブジェクトネットワーク130および汎用ラッパープログラム121を介して処理を要求し、その実行結果を利用するような、ある特定の処理を実行する。汎用ラッパープログラム121は、定義情報格納部141に格納されている情報に従って、オブジェクト管理サービス151に、サーバアプリケーション用のオブジェクトを登録し、また、クライアントアプリケーション111からの入力情報をサーバアプリケーション101の入力形式に変換して転送し、また、サーバアプリケーション101からの出力情報をクライアントアプリケーション111に渡すべき出力形式に変換して転送する。

【0022】定義情報格納部141は、サーバアプリケーション101の入出力とクライアントアプリケーション111の入出力との間のデータ変換を定義する情報、汎用ラッパープログラム121と、サーバアプリケーション101およびクライアントアプリケーション111との間の通信に関する定義情報など、汎用ラッパープログラム121がラッピングする際に必要とする、サーバアプリケーション101およびクライアントアプリケーション111に固有な各種情報を格納する。オブジェクト管理サービス151は、分散オブジェクト環境上で使用できるオブジェクトの識別子や存在場所などの情報を管理し、分散オブジェクト環境内の各アプリケーションからの要求に応じて、分散オブジェクト環境で利用可能なオブジェクトの登録、削除、検索を行うことができる。分散オブジェクト環境内のアプリケーションが、オブジェクト管理サービス151にオブジェクトを登録することによって、その登録されたオブジェクトは分散オブジェクト環境内で利用可能な状態となり、また、分散オブジェクト環境内のアプリケーションが、利用したいオブジェクトの情報をオブジェクト管理サービス151に問い合わせ、情報を取得することで、そのアプリケーションは利用したいオブジェクトを呼び出すことができ

る。

【0023】図2、図3は、図1に示す本発明の一実施形態の動作を説明するために引用した図であり、それぞれ、制御手順をフローチャートで、動作概念をブロックで示した図である。以下、図2、図3を参照しながら図1に示す本発明一実施形態の動作について説明する。

【0024】まず、汎用ラッパープログラム121は、オブジェクト管理サービス151に、サーバアプリケーション用のオブジェクトを登録しておく（図2のステップA1）。クライアントアプリケーションは、処理を要求するサーバアプリケーション101に対応するオブジェクト情報を、オブジェクト管理サービス151に問い合わせることによって、サーバアプリケーション用のオブジェクトの情報を取得する（ステップA2）。クライアントアプリケーション111は、取得した情報を用いて、サーバアプリケーション用のオブジェクトを呼び出すことで、汎用ラッパープログラム121にサーバアプリケーション101へ渡すためのデータが送信される（ステップA3）。

【0025】汎用ラッパープログラム121は、クライアントアプリケーション111からの入力データを受信し（ステップA4）、定義情報格納部141上の情報に基づいて、サーバアプリケーション101の入力形式にデータを変換する（ステップA5）。次に、汎用ラッパープログラム121は、変換した入力データを、サーバアプリケーション101のインターフェースに則った形式で送信し（ステップA6）、サーバアプリケーション101はこれを受信する（ステップA7）。入力データを受け取ったサーバアプリケーション101は、その入力データに基づいてプログラムを実行し（ステップA8）、実行結果を含む出力データを送信する（ステップA9）。

【0026】汎用ラッパープログラム121は、サーバアプリケーション101の出力データを受信し（ステップA10）、定義情報格納部141上の情報に基づいて、クライアントアプリケーション111が要求する出力データに変換する（ステップA11）。最後に、汎用ラッパープログラム121は変換した出力データを送信し（ステップA12）、クライアントアプリケーション111はこれを受信する（ステップA13）ことで、クライアントアプリケーション111からのサーバアプリケーション101の実行は完了する。

【0027】なお、上述した本発明実施形態において、汎用ラッパープログラム121そのものの動作は従来のラッパープログラム（図8、122）と変わらないが、サーバアプリケーション101の入出力とクライアントアプリケーション111が必要とする入出力とのデータ変換や、サーバアプリケーション101と通信するために必要な情報、オブジェクト管理サービス151に登録する情報など、クライアントアプリケーション111とサ

サーバアプリケーション101との間のやりとりに必要な情報に関して、汎用ラッパープログラム121自身が定義情報格納部141に格納されている情報を参照することにより、このため、サーバアプリケーション101やクライアントアプリケーション111の仕様に変更が生じたときに、定義情報格納部141に格納されている情報を変更するだけで、汎用ラッパープログラム121の動作を変更することができる。

【0028】すなわち、汎用ラッパープログラム121自身には全く手を加えることなく、サーバアプリケーション101やクライアントアプリケーション111の変更を汎用ラッパープログラム121に反映させることができる。また、上述した本発明実施形態では、さらに、サーバコンピュータ100、クライアントコンピュータ110、通信管理装置120、オブジェクト管理サーバコンピュータ150それぞれへの通信に関する定義情報が変更された場合も、汎用ラッパープログラム121自身には手を加えることなく通信可能にすることができる。それは、定義情報格納部141に、分散オブジェクト環境側および分散オブジェクト環境側それぞれの通信に関する定義情報も含まれており、汎用ラッパープログラム121は、その情報を参照しながら通信手段を生成しているためである。

【0029】次に、具体例を用いて本発明実施形態の動作を詳細に説明する。図3に示すように、例えば、分散オブジェクト環境内の電文の形式が、電文1、12のように、上下方向とも、インターフェース名、オペレーション名、入出力の属性、型、値、属性、型、値、…という形式になっており、サーバアプリケーション用の電文は、上り電文は、電文21のように、関数名、入力値、入力値、…という形式で、下り電文が、電文22のように、関数名、出力値、出力値、…という形式であるとする。このとき、汎用ラッパープログラム121は、分散オブジェクト環境内上り電文11をサーバアプリケーション用上り電文21に、サーバアプリケーション用下り電文22を分散オブジェクト環境内下り電文12に変換する。電文11と電文21の間および電文21と電文22の間の変換規則を定義するのが、インターフェース定義情報31であり、定義情報格納部141に格納されている。そして、このインターフェース定義情報31が汎用ラッパープログラム121に反映される。

【0030】図3に示す例では、インターフェース定義情報31は、クライアント側メソッド名、クライアント側オペレーション名、クライアント側入力情報、クライアント側出力情報、サーバ側関数名、サーバ側入力情報、サーバ側出力情報の順に記述しており、クライアント側の情報とサーバ側の情報との間は“:”で区切っており、クライアント情報、サーバ情報ともに、識別名と、入力情報と出力情報との間は“:”で区切っている。また、各情報に要素が複数ある場合は、その間

は“,”で区切っている。

【0031】ここで、インターフェース定義情報31における定義例について、図3のインターフェース定義情報31の1行目を例に挙げて説明する。クライアントアプリケーション111で呼ぶ分散オブジェクトのインターフェース名がn11_Aであり、オペレーション名がMtd_Aであった場合は、サーバアプリケーション101内のTx_Aを呼び出せることを意味しており、また、クライアントからの2つの入力データ型がともにlong型であり、それをサーバアプリケーション101の引数にするときにはともにデータ型をINT4型に変換すること、また、サーバ側の出力データ型がINT4型であり、それをクライアントへの出力にする時には、データ型をlong型に変換することを意味している。

【0032】また、クライアントアプリケーション111と汎用ラッパープログラム121の間、および、汎用ラッパープログラム121とサーバアプリケーション101の間で通信を行うために必要な定義情報は、通信定義情報32として、定義情報格納部141に格納されている。図3に示すのは、通信定義情報32の1行目でオブジェクト管理サーバコンピュータのIPアドレスを、2行目でサーバアプリケーション101が存在するサーバコンピュータの識別名を定義している。インターフェース定義情報31および通信定義情報32を汎用ラッパープログラム121が参照することで、汎用ラッパープログラム121はオブジェクト管理サービス151へのオブジェクトの登録および、クライアントとサーバとの間やラッパーとサーバとの間の通信と電文の変換を行うことができる。

【0033】図4は本発明が採用されるラッピングシステムの他の実施形態(第2の実施形態)を示すブロック図である。図1に示す第1の実施形態との差異は、通信管理装置170が、図1に示す通信管理装置120の構成に加え、IDL(Interface Definition Language)生成手段171を有することである。IDL生成手段171は、定義情報格納部141に格納されているインターフェース定義情報を参照することで、分散オブジェクト環境上のインターフェースを定義するための情報であるIDLを生成する。なお、クライアントアプリケーション111を作成する場合は、IDLを翻訳すること、クライアントアプリケーションのインターフェース部分に相当するソースコードを生成することができ、そのソースコードを基にして作成することができる。ここでは、分散オブジェクト環境上でのインターフェース定義を自動的に作成することによって、クライアントアプリケーション111を作成する手間を軽減することができる。

【0034】なお、上述した図4に示す第2の実施形態では、通信管理装置170上でIDL生成手段171が

13

動作する場合について説明したが、I DL 生成手段171は、図1における定義情報格納部141内の情報を参照することができる任意のコンピュータ上で動作させてもよい。また、図1、図4に示すそれぞれ第1および第2の実施の形態では、通信管理装置120上で汎用ラッパープログラム121が動作する場合について説明したが、図5に示すように、汎用ラッパープログラム121をサーバコンピュータ160上で動作させることもでき、また、図6に示すようにクライアントコンピュータ110上で動作させてもよい。

【0035】更に、図1、図4に示す第1および第2の実施の形態では、分散オブジェクト環境側に存在するクライアントアプリセッションが非分散オブジェクト環境上に存在するサーバアプリセッションに対して処理を要求する形態について説明したが、非分散オブジェクト環境側に存在するアプリセッションが分散オブジェクト環境側に存在するアプリセッションに対して処理を要求する場合や、また、分散オブジェクト環境上に存在するアプリセッションと非分散オブジェクト環境上に存在するアプリセッションが相互に処理を要求する場合も同様である。また、第1および第2の実施の形態では、オブジェクト管理サーバコンピュータ150上でオブジェクト管理サービス151が動作する場合について説明したが、オブジェクト管理サービス151は、図1におけるクライアントコンピュータ110や通信管理装置120や図5におけるサーバコンピュータ160を含めて分散オブジェクト環境上に存在する任意のコンピュータ上で動作させてもよい。

【0036】図7は、本発明が採用されるラッピングシステムの更に他の実施形態（第3の実施形態）を示すブロック図である。図7を参照すると、本発明における第3の実施の形態は、汎用ラッパープログラムを記録した記録媒体400を備える。この記録媒体400は磁気ディスク、半導体メモリその他の記録媒体であってよい。汎用ラッパープログラムは記録媒体400から通信管理装置300に読み込まれ、通信管理装置300の動作を制御する。通信管理装置300は、汎用ラッパープログラムの制御により以下の処理、すなわち第1および第2の実施形態における通信管理装置120および170による処理と同様の処理、を実行する。

【0037】まず、汎用ラッパープログラムは、オブジェクト管理サービス151に、サーバアプリセッション用のオブジェクトを登録しておく。クライアントアプリセッションは、処理を要求するサーバアプリセッション101に対応するオブジェクト情報を、オブジェクト管理サービス151に問い合わせることによって、サーバアプリセッション用のオブジェクトの情報を取得する。クライアントアプリセッション111は、取得した情報を用いてサーバアプリセッション用のオブジェクトを呼び出すことで、汎用ラッパープログラムにサーバアプリ

14

セッション101へ渡すためのデータが送信される。次に、汎用ラッパープログラムは、クライアントアプリセッション111からの入力データを受信し、定義情報格納部141上の情報に基づいて、サーバアプリセッション101の入力形式にデータを変換する。そして、汎用ラッパープログラムは、変換した入力データを、サーバアプリセッション101のインターフェースに則した形式で送信し、サーバアプリセッション101はこれを受信する。

【0038】入力データを受け取ったサーバアプリセッション101は、その入力データに基づいてプログラムを実行し、実行結果を含む出力データを送信する。汎用ラッパープログラムは、サーバアプリセッション101の出力データを受信し、定義情報格納部141上の情報に基づいて、クライアントアプリセッション111が要求する出力データに変換する。最後に、汎用ラッパープログラムは変換した出力データを送信し、クライアントアプリセッション111はこれを受信することで、クライアントアプリセッション111からのサーバアプリセッション101の実行は完了する。また、記録媒体400に記憶されている汎用ラッパープログラムは、I DL 生成手段171が含まれている場合には、I DL 生成手段171は、定義情報格納部141に格納されているインターフェース定義情報を参照することで、分散オブジェクト環境上のインターフェースを定義するための情報であるI DL を生成する。なお、クライアントアプリセッション111を作成する場合は、このI DL の解釈手法を用いることで、クライアントアプリセッションのインターフェース部分に相当するソースコードを生成することで、そのソースコードを基にして作成することができる。

【0039】図1、図4および図7に示す、第1、第2および第3の実施の形態では、処理の受け渡しを行うサーバアプリセッションとクライアントアプリセッションの対応が1対1の場合について説明したが、処理の受け渡しを行うサーバアプリセッションとクライアントアプリセッションの対応が1対多、多対1、多対多でよく、その数に制限はない。また、この場合、サーバアプリセッションが動作するサーバコンピュータの数およびクライアントアプリセッションが動作するクライアントコンピュータの数も、それぞれ複数でもよく、その数に制限はない。

【0040】

【発明の効果】以上説明のように本発明によれば、ラッパープログラムがインターフェースの定義情報を参照して動作するため、その定義情報を変更するだけで、インターフェースの変更がラッパープログラムに反映される。したがって、非分散オブジェクト環境内のアプリセッションのインターフェースに変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、ラッパ

一プログラムに対してインターフェースの変更を反映できる。このため、非分散オブジェクト環境内のアプリケーションのインターフェースに変更が生じた際にプログラムの負担が大幅に軽減される。

【0041】また、同理由で分散オブジェクト環境内のアプリケーションのインターフェースに変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、ラッパープログラムに対してインターフェースの変更を反映できる。このため、分散オブジェクト環境内のアプリケーションのインターフェースに変更が生じた際にプログラムの負担が大幅に軽減される。

【0042】更に、ラッパープログラムが分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義情報を参照して動作するため、その定義情報を変更するだけで、通信定義の変更がラッパープログラムに反映されことから、分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義に変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、ラッパープログラムに対して通信定義の変更を反映できる。このため、分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義に変更が生じた際にプログラムの負担が大幅に軽減される。

【0043】また、定義情報格納部に格納されているインターフェース情報から、分散オブジェクト環境上でのインターフェース定義であるIDLを自動的に作成することによって、非分散オブジェクト環境上のアプリケーションを利用する分散オブジェクト環境内のクライアントアプリケーションを作成する手間を軽減することができることもできる。

【図面の簡単な説明】

【図1】本発明における第1の実施形態の構成を示すブロック図である。

【図2】図1に示す実施形態の動作を示すフローチャートである。

【図3】図1に示す実施形態の動作の具体例を示す図である。

【図4】本発明における第2の実施形態の構成を示すブロック図である。

【図5】図1に示す第1の実施形態において、汎用ラッパープログラムがサーバコンピュータに含まれる場合の構成を示すブロック図である。

【図6】図1に示す第1の実施形態において、汎用ラッパープログラムがクライアントコンピュータに含まれる場合の構成を示すブロック図である。

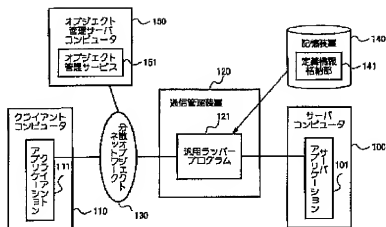
【図7】本発明における第3の実施形態の構成を示すブロック図である。

【図8】従来のラッピングシステムの構成例を示すブロック図である。

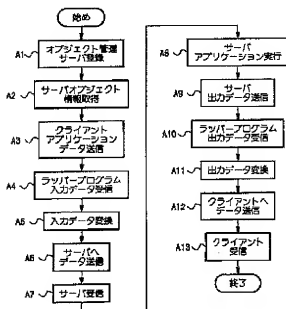
【符号の説明】

- 11 分散オブジェクト環境内上り電文
- 12 分散オブジェクト環境内下り電文
- 21 サーバアプリケーション用上り電文
- 22 サーバアプリケーション用下り電文
- 31 インターフェース定義情報
- 32 通信定義情報
- 100、160 サーバコンピュータ
- 101 サーバアプリケーション
- 110、180 クライアントコンピュータ
- 111 クライアントアプリケーション
- 120、170、300 通信管理装置
- 121 汎用ラッパープログラム
- 122 ラッパープログラム
- 123 ラッパーソースコード生成手段
- 124 ラッパーソースコード
- 125 コンパイラ
- 130 分散オブジェクトネットワーク
- 140 記憶装置
- 141 定義情報格納部
- 150 オブジェクト管理サーバコンピュータ
- 151 オブジェクト管理サービス
- 171 IDL生成手段
- 400 記録媒体

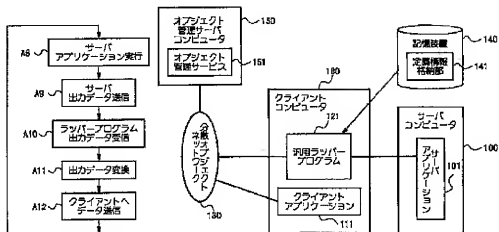
【 図1 】



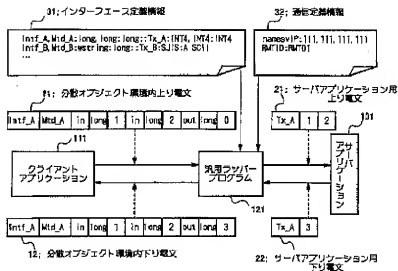
【 図2 】



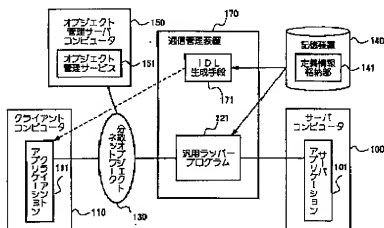
【 図6 】



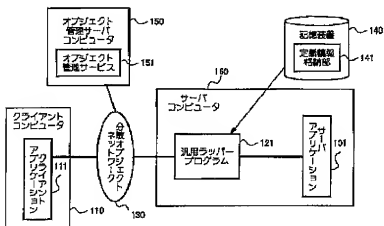
【 図3 】



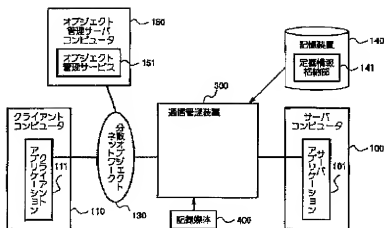
【 図4 】



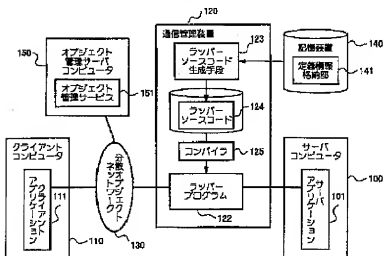
【 図5 】



【 図7 】



【 図8 】



PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-337935

(43)Date of publication of application : 07.12.2001

(51)Int.Cl.

G06F 15/16

G06F 9/44

(21)Application number : 2000-153678

(71)Applicant : NEC CORP

(22)Date of filing : 24.05.2000

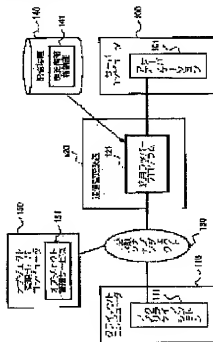
(72)Inventor : NISHIDA TAKESHI

(54) APPLICATION WRAPPING METHOD IN DISPERSED OBJECT ENVIRONMENT, ITS COMMUNICATION MANAGEMENT DEVICE, AND RECORDING MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To update wrapper program operation without any alteration of a wrapper program itself even when definition information necessary for wrapping is changed in wrapping over a dispersed object environment of application on a non-dispersed object environment.

SOLUTION: When a general-purpose wrapper program 121 refers to a definition information storage part 141 storing definition information necessary for wrapping, operation of the general-purpose wrapper program 121 is decided. When information necessary for the wrapping such as server application 101 to be wrapped and an interface for a client application 111 is changed, the definition information in the definition information storage part 141 is changed without changing any general-purpose wrapper program 121 itself, so that operation of the general-purpose wrapper program 121 can be changed.



LEGAL STATUS

[Date of request for examination] 13.04.2001

[Date of sending the examiner's decision of rejection] 12.11.2002

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-337935

(P2001-337935A)

(43) 公開日 平成13年12月7日 (2001.12.7)

(51) IntCl. ⁷	識別記号	F I	テーマコード(参考)
G 0 6 F 15/16	6 2 0	G 0 6 F 15/16	6 2 0 T 5 B 0 4 5
9/44		9/06	6 2 0 A 5 B 0 7 6

審査請求 有 請求項の数 9 O L (全 12 頁)

(21) 出願番号 特願2000-153678(P2000-153678)

(22) 出願日 平成12年5月24日 (2000.5.24)

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 西田 武史

東京都港区芝五丁目7番1号 日本電気株式会社内

(74) 代理人 100108578

弁理士 高橋 昭男 (外3名)

Fターム(参考) 5B045 G001

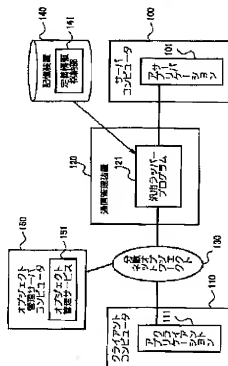
5B076 DF06

(54) 【発明の名称】 分散オブジェクト環境におけるアプリケーションのラッピング方法、その通信管理装置及び記録媒体

(57) 【要約】

【課題】 非分散オブジェクト環境上のアプリケーションの分散オブジェクト環境へのラッピングについて、ラッピングに必要な定義情報に変更が生じても、ラッパープログラム自身に手を加えることなく、ラッパープログラムの動作を更新できるようにする。

【解決手段】 ラッピングに必要な定義情報を格納している定義情報格納部141を汎用ラッパープログラム121が参照することで汎用ラッパープログラム121の動作を決定する。ラッピングするサーバアプリケーション101やクライアントアプリケーション111のインターフェース等、ラッピングに必要な情報に変更が生じたときには、汎用ラッパープログラム121自身には手を加えずに、定義情報格納部141上の定義情報を変更することによって、汎用ラッパープログラム121の動作を変更することができる。



【特許請求の範囲】

【請求項1】 非分散オブジェクト環境上に存在する少なくとも1個のアプリケーションを分散オブジェクト環境上のオブジェクトとして利用できるように、前記それぞれのアプリケーション間で処理の受け渡しを行う分散オブジェクト環境におけるアプリケーションのラッピング方法であって、前記非分散オブジェクト環境上に存在するアプリケーションを前記分散オブジェクト環境上のオブジェクトとしてラッピングするために必要な情報を参照することによって前記非分散オブジェクト環境上に存在するアプリケーションのラッピングの方法を決定し、その方法に従って動作を決定することを特徴とする分散オブジェクト環境におけるアプリケーションのラッピング方法。

【請求項2】 前記ラッピングは、前記分散オブジェクト環境上の利用可能オブジェクトを管理しているオブジェクト管理サービスに対して前記非分散オブジェクト環境上に存在するアプリケーションの登録もしくは削除を行い、前記分散オブジェクト環境上に存在するアプリケーションから送られてきた電文の形式を前記分散オブジェクト環境上に存在するアプリケーションへ送る電文形式に変換し、前記非分散オブジェクト環境上に存在するアプリケーションから送られてきた電文の形式を前記分散オブジェクト環境上に存在するアプリケーションへ送る電文形式に変換することを特徴とする請求項1に記載の分散オブジェクト環境におけるアプリケーションのラッピング方法。

【請求項3】 更に、前記ラッピングに必要な情報から前記分散オブジェクト環境上に存在するアプリケーションの作成を支援するインターフェース定義情報を作成することを特徴とする請求項1または請求項2に記載の分散オブジェクト環境におけるアプリケーションのラッピング方法。

【請求項4】 非分散オブジェクト環境上に存在する少なくとも1個のアプリケーションを分散オブジェクト環境上のオブジェクトとして利用できるように、前記それぞれのアプリケーション間で処理の受け渡しを行う通信管理装置であって、前記非分散オブジェクト環境上に存在するアプリケーションを分散オブジェクト環境上に存在するオブジェクトとしてラッピングするために必要な情報を参照することによって前記非分散オブジェクト環境上に存在するアプリケーションのラッピングの方法を決定し、その方法に従って動作を決定するラッピング手段を備えたことを特徴とする通信管理装置。

【請求項5】 前記ラッピング手段は、前記分散オブジェクト環境上に存在する利用可能オブジェクトを管理しているオブジェクト管理サービスに対して前記非分散オブジェクト環境上に存在するアプリケーションの登録もしくは削除をする手段と、前記分散オブジェクト環境上に存在するアプリケーションから送られてきた電文の形

式を前記非分散オブジェクト環境上に存在するアプリケーションへ送る電文形式に変換する手段と、前記非分散オブジェクト環境上にあるアプリケーションから送られてきた電文の形式を前記分散オブジェクト環境上に存在するアプリケーションへ送る電文形式に変換する手段とを含むことを特徴とする請求項4に記載の通信管理装置。

【請求項6】 前記ラッピング手段は、前記ラッピングに必要な情報から分散オブジェクト環境上に存在するアプリケーションの作成を支援するインターフェース定義情報を作成する手段を更に含むことを特徴とする請求項4または請求項5に記載の通信管理装置。

【請求項7】 非分散オブジェクト環境上に存在する少なくとも1組のアプリケーションを分散オブジェクト環境上のオブジェクトとして利用できるように、前記それぞれのアプリケーション間で処理の受け渡しを行う、分散オブジェクト環境におけるアプリケーションのラッピング処理を行うプログラムであって、前記非分散オブジェクト環境上に存在するアプリケーションを前記分散オブジェクト環境上のオブジェクトとしてラッピングするために必要な情報を参照するステップと、前記参照ステップによって前記非分散オブジェクト環境上に存在するアプリケーションのラッピング方法を決定し、それに従って動作を決定するステップとがプログラムされ記録されたコンピュータ読み取り可能な記録媒体。

【請求項8】 前記ラッピング方法を決定し、それに従って動作を決定するステップは、前記分散オブジェクト環境上の利用可能オブジェクトを管理しているオブジェクト管理サービスに対して前記非分散オブジェクト環境上にあるアプリケーションの登録もしくは削除を行うステップと、前記分散オブジェクト環境上のアプリケーションから送られてきた電文の形式を前記非分散オブジェクト環境上にあるアプリケーションへ送る電文形式に変換するステップと、前記非分散オブジェクト環境上にあるアプリケーションから送られてきた電文の形式を前記分散オブジェクト環境上のアプリケーションへ送る電文形式に変換するステップとを含むことを特徴とするコンピュータ読み取り可能な請求項7に記載の記録媒体。

【請求項9】 前記ラッピング方法を決定し、それに従って動作を決定するステップは、更に、ラッピングに必要な情報から分散オブジェクト環境上のアプリケーションの作成を支援するためのインターフェース定義情報を作成するステップを含むことを特徴とするコンピュータ読み取り可能な請求項7または請求項8に記載の記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、分散オブジェクト環境におけるアプリケーションのラッピング方法ならびにその通信管理装置および同方法がプログラムされ記録

された記録媒体に關し、特に、非分散オブジェクト環境上に存在するアプリケーションと分散オブジェクト環境上のアプリケーションとの間で、一方が他方に処理を要求し、処理を要求されたアプリケーションが処理を要求したアプリケーションに処理結果を返すことのできる、分散オブジェクト環境におけるアプリケーションのラッピング方法ならびにその通信管理装置および同方法がプログラムされ記録された記録媒体に關する。

【0002】

【従来の技術】OMG(Object Management Group)が提唱しているCORBA(Common Object Request Broker Architecture)に代表される分散オブジェクト環境において、例えば汎用コンピュータ上のトランザクションシステムのような既存の非分散オブジェクト環境のアプリケーションを分散オブジェクト環境上で利用する方法として、非分散オブジェクト環境のアプリケーションへの入出力を分散オブジェクト環境上のインターフェースに変換する方法がある。上記のようにして、非分散オブジェクト環境上のアプリケーションを分散オブジェクト環境上のオブジェクトにみせかけることをラッピングといい、また、これを実現する機能部をラッパーという。

【0003】ラッピングシステムの一例を図8に示す。図に示すように、ラッピングシステムは、サーバコンピュータ100と、クライアントコンピュータ110と、通信管理装置120と、分散オブジェクトネットワーク130と、記憶装置140と、オブジェクト管理サーバコンピュータ150とから構成されている。サーバコンピュータ100は、サーバアプリケーション101を備えている。サーバアプリケーション101は、サーバコンピュータ100の外部への入出力インターフェースを持ち、外部からの要求に応じて特定の処理を実行し、その実行結果を返すプログラムである。クライアントコンピュータ110は、クライアントアプリケーション111を備えている。クライアントアプリケーション111は、クライアントコンピュータ110の外部への入出力インターフェースを持ち、サーバアプリケーション101に処理を要求し、その結果を反映するような、ある特定の処理を行うプログラムである。

【0004】通信管理装置120は、ラッパープログラム122を備えている。ラッパープログラム122は、ラッパーソースコード生成手段123およびコンパイラ125によって生成され、クライアントアプリケーション111のインターフェースとサーバアプリケーション101のインターフェースとの間の変換と、通信管理を行うプログラムである。記憶装置140は、定義情報格納部141を備えている。定義情報格納部141は、サーバアプリケーション101のインターフェース情報などといった、非分散オブジェクト環境上のアプリケーションに対するラッピングに必要な定義情報を格納してい

る機能部である。

【0005】オブジェクト管理サーバコンピュータ150は、オブジェクト管理サービス151を備えている。オブジェクト管理サービス151は、分散オブジェクト環境上のオブジェクトを一意に識別するための情報を管理するプログラムで、クライアントプログラムは、オブジェクト管理サービス151に問い合わせることによって、オブジェクトを一意に識別するための情報を取得することができ、この情報をもちいることで、目的のオブジェクトに処理を要求することができる。

【0006】このような構成を有する従来のラッピングシステムは次のように動作する。まず、ラッパーソースコード生成手段123は、定義情報格納部141に格納されている定義情報を参照して、ラッパーソースコード124を生成する。そして、ラッパーソースコード124をコンパイラ125でコンパイル、リンクすることで、ラッパープログラム122を生成する。

【0007】生成されたラッパープログラム122は、サーバアプリケーション101用のオブジェクトをオブジェクト管理サービス151に登録する。クライアントアプリケーション111は、オブジェクト管理サービス151にあるオブジェクト情報を参照することで、ラッパープログラム122と通信を行う。クライアントアプリケーション111から入力データを受け取ったラッパープログラム122は、その入力データをサーバアプリケーション101の入力形式に変換して、サーバアプリケーション101に渡す。サーバアプリケーション101がラッパープログラム122に出力データを返すと、ラッパープログラム122は、その出力データをクライアントアプリケーション111が要求している形式に変換して返す。

【0008】以上の処理を行うことで、分散オブジェクト環境上にあるクライアントアプリケーション111から非分散オブジェクト環境下のサーバアプリケーション101を利用することが可能になる。

【0009】

【発明が解決しようとする課題】上述した従来技術によれば、ラッピング対象となる非分散オブジェクト環境上のアプリケーションのインターフェースや、非分散オブジェクト環境上のアプリケーションとの処理の受け渡しを行う分散オブジェクト環境上のアプリケーション、非分散オブジェクト環境や分散オブジェクト環境上の通信に關する定義など、ラッパーが必要とする定義情報が1つでも変化するとき、ラッパープログラムそのものも再度生成しなおさなければならない。すなわち、従来のラッピング方法によれば、ラッピングに必要な各定義情報を基にして、それに対応したラッパープログラムのソースコードを生成し、それをコンパイル、リンクするという方式をとっていたため、ラッパーが必要とする定義情報が1つでも変化するとき、ラッパープログラムそ

のものも再度生成し直す必要がある。したがって、プログラマに対する負担が大きく、効率が悪かった。

【0010】本発明は上記事情に鑑みてなされたものであり、非分散オブジェクト環境上に存在するアプリケーションを前記分散オブジェクト環境上のオブジェクトとしてラッピングするために必要な情報を参照することによって前記非分散オブジェクト環境上に存在するアプリケーションのラッピングの方法を決定し、その方法に従って動作を決定することにより、ラッピングに必要な定義情報に変更が生じても、ラッパープログラムのオブジェクトコードを再度生成することなしに、ラッパープログラムの動作を更新することができる、分散オブジェクト環境におけるアプリケーションのラッピング方法ならびにその通信管理装置および前記方法がプログラムされ記録された記録媒体を提供することを目的とする。

【0011】

【課題を解決するための手段】上述した課題を解決するために請求項1に記載の発明は、非分散オブジェクト環境上に存在する少なくとも1個のアプリケーションを分散オブジェクト環境上のオブジェクトとして利用できるように、それぞれのアプリケーション間で処理の受け渡しを行う分散オブジェクト環境におけるアプリケーションのラッピング方法であって、非分散オブジェクト環境上に存在するアプリケーションを分散オブジェクト環境上のオブジェクトとしてラッピングするために必要な情報を参照することによって非分散オブジェクト環境上に存在するアプリケーションのラッピングの方法を決定し、その方法に従って動作を決定することとした。このことにより、ラッパープログラムがインターフェースの定義情報を参照して動作するため、その定義情報を変更するだけで、インターフェースの変更がラッパープログラムに反映され、したがって、非分散もしくは分散オブジェクト環境内のアプリケーションのインターフェースに変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、ラッパープログラムに対してインターフェースの変更を反映できる。

【0012】請求項2に記載の発明は、請求項1に記載の分散オブジェクト環境におけるアプリケーションのラッピング方法において、ラッピングは、分散オブジェクト環境上の利用可能オブジェクトを管理しているオブジェクト管理サービスに対して非分散オブジェクト環境上に存在するアプリケーションの登録もしくは削除を行い、分散オブジェクト環境上に存在するアプリケーションから送られてきた電文の形式を前記非分散オブジェクト環境上に存在するアプリケーションへ送る電文形式に変換し、非分散オブジェクト環境上に存在するアプリケーションから送られてきた電文の形式を前記分散オブジェクト環境上に存在するアプリケーションへ送る電文形式に変換することとした。このことにより、ラッパープログラムが分散オブジェクト環境内および非分散オブ

ジェクト環境内の通信に関する定義情報を参照して動作するため、その定義情報を変更するだけで、通信定義の変更がラッパープログラムに反映され、したがって、分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義に変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、ラッパープログラムに対して通信定義の変更を反映できる。

【0013】請求項3に記載の発明は、請求項1または請求項2に記載の分散オブジェクト環境におけるアプリケーションのラッピング方法において、更に、ラッピングに必要な情報から分散オブジェクト環境上に存在するアプリケーションの作成を支援するインターフェース定義情報を作成することとした。このことにより、あらかじめ格納されているインターフェース情報から、分散オブジェクト環境上でのインターフェース定義である、例えばDLを自動的に作成することによって、非分散オブジェクト環境上のアプリケーションを利用する分散オブジェクト環境内のクライアントアプリケーションを作成する手間を軽減することができることもできる。

【0014】請求項4に記載の発明は、非分散オブジェクト環境上に存在する少なくとも1個のアプリケーションを分散オブジェクト環境上のオブジェクトとして利用できるように、それぞれのアプリケーション間で処理の受け渡しを行う通信管理装置であって、非分散オブジェクト環境上に存在するアプリケーションを分散オブジェクト環境上に存在するオブジェクトとしてラッピングするために必要な情報を参照することによって非分散オブジェクト環境上に存在するアプリケーションのラッピング方法を決定し、その方法に従って動作を決定するラッピング手段を備えることとした。上記構成により、ラッパープログラムがインターフェースの定義情報を参照して動作するため、その定義情報を変更するだけで、インターフェースの変更がラッパープログラムに反映され、したがって、非分散もしくは分散オブジェクト環境内のアプリケーションのインターフェースに変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、ラッパープログラムに対してインターフェースの変更を反映できる。このため、非分散もしくは分散オブジェクト環境内のアプリケーションのインターフェースに変更が生じた際にプログラマの負担を大幅に軽減したような新管理装置を提供でき、効率のよいラッピングシステムを構築できる。

【0015】請求項5に記載の発明は、請求項4に記載の通信管理装置において、ラッピング手段は、分散オブジェクト環境上に存在する利用可能オブジェクトを管理しているオブジェクト管理サービスに対して前記非分散オブジェクト環境上に存在するアプリケーションの登録もしくは削除をする手段と、分散オブジェクト環境上に存在するアプリケーションから送られてきた電文の形式を前記非分散オブジェクト環境上に存在するアプリケー

ショーへ送る電文形式に変換する手段と、非分散オブジェクト環境上にあるアプリケーションから送られてきた電文の形式を前記分散オブジェクト環境上に存在するアプリケーションへ送る電文形式に変換する手段を含むこととした。上記構成により、ラッパープログラムが分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義情報を参照して動作するため、その定義情報を変更するだけで、通信定義の変更がラッパープログラムに反映されことから、分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義に変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、ラッパープログラムに対して通信定義の変更を反映できる。このため、分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義に変更が生じた際にプログラムの負担を大幅に軽減した通信管理装置を提供でき、効率のよいラッピングシステムを構築できる。

【0016】請求項6に記載の発明は、請求項4または請求項5に記載の通信管理装置において、ラッピング手段は、ラッピングに必要な情報から分散オブジェクト環境上に存在するアプリケーションの作成を支援するインターフェース定義情報を作成する手段を更に含むこととした。上記構成により、あらかじめ格納されているインターフェース情報から、分散オブジェクト環境上でのインターフェース定義である、例えばIDLを自動的に作成することによって、非分散オブジェクト環境上のアプリケーションを利用する分散オブジェクト環境内のクライアントアプリケーションを作成する手間を軽減した通信管理装置を提供でき、また、効率のよいラッピングシステムを構築できる。

【0017】請求項7に記載の発明は、非分散オブジェクト環境上に存在する少なくとも1個のアプリケーションを分散オブジェクト環境上のオブジェクトとして利用できるように、それぞれのアプリケーション間で処理の受け渡しを行う、分散オブジェクト環境におけるアプリケーションのラッピング処理を行うプログラムであって、非分散オブジェクト環境上に存在するアプリケーションを前記分散オブジェクト環境上のオブジェクトとしてラッピングするために必要な情報を参照するステップと、参照ステップによって前記分散オブジェクト環境上に存在するアプリケーションのラッピング方法を決定し、それによって動作を決定するステップとがプログラムされ記録されることとした。上記したプログラムをコンピュータが読み出し実行することにより、ラッパープログラムがインターフェースの定義情報を参照して動作するため、その定義情報を変更するだけで、インターフェースの変更がラッパープログラムに反映され、したがって、非分散もしくは分散オブジェクト環境内のアプリケーションのインターフェースに変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、

ラッパープログラムに対してインターフェースの変更を反映できる。このため、分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義に変更が生じた際にプログラムの負担を大幅に軽減することができる。

【0018】請求項8に記載の発明は、請求項7に記載の記録媒体において、ラッピング方法を決定し、それによって動作を決定するステップは、分散オブジェクト環境上の利用可能なオブジェクトを管理しているオブジェクト管理サービスに対して非分散オブジェクト環境上にあるアプリケーションの登録もしくは削除を行うステップと、分散オブジェクト環境上のアプリケーションから送られてきた電文の形式を分散オブジェクト環境上にあるアプリケーションへ送る電文形式に変換するステップと、非分散オブジェクト環境上にあるアプリケーションから送られてきた電文の形式を分散オブジェクト環境上のアプリケーションへ送る電文形式に変換するステップとを含むこととした。上記プログラムをコンピュータが読み出し実行することにより、ラッパープログラムが分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義情報を参照して動作するため、その定義情報を変更するだけで、通信定義の変更がラッパープログラムに反映されことから、分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義に変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、ラッパープログラムに対して通信定義の変更を反映できる。このため、分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義に変更が生じた際にプログラムの負担を大幅に軽減することができる。

【0019】請求項9に記載の発明は、請求項7または請求項8に記載の記録媒体において、ラッピング方法を決定し、それによって動作を決定するステップは、更に、ラッピングに必要な情報から分散オブジェクト環境上のアプリケーションの作成を支援するためのインターフェース定義情報を作成するステップを含むこととした。上記プログラムをコンピュータが読み出し実行することにより、あらかじめ格納されているインターフェース情報から、分散オブジェクト環境上でのインターフェース定義である、例えばIDLを自動的に作成することによって、非分散オブジェクト環境上のアプリケーションを利用する分散オブジェクト環境内のクライアントアプリケーションを作成する手間を軽減することができる。

【0020】

【発明の実施形態】以下、図面を使用して本発明の実施形態について詳細に説明する。図1は本発明が採用されるラッピングシステムの一実施形態を示すブロック図である。図1において、ラッピングシステムは、プログラム制御により動作するサーバコンピュータ100と、クライアントコンピュータ110と、通信管理装置120

と、分散オブジェクトネットワーク130、記憶装置140、オブジェクト管理サーバコンピュータ150とから構成されている。サーバコンピュータ100は、サーバアプリケーション101を含み、クライアントコンピュータ110は、クライアントアプリケーション111を含む。また、通信管理装置120は、汎用ラッパープログラム121を含み、記憶装置140は、定義情報格納部141を含む。更に、オブジェクト管理サーバコンピュータ150は、オブジェクト管理サービス151を含む。

【0021】サーバアプリケーション101は、サーバコンピュータ100の外部との入力インターフェースもしくは入力インターフェースと出力インターフェースの両方を持ち、外部からの要求に応じてある特定の処理を実行する。クライアントアプリケーション111は、クライアントコンピュータ110上で動作し、サーバアプリケーション101に対して分散オブジェクトネットワーク130および汎用ラッパープログラム121を介して処理を要求し、その実行結果を利用するよう、ある特定の処理を実行する。汎用ラッパープログラム121は、定義情報格納部141に格納されている情報に従って、オブジェクト管理サービス151に、サーバアプリケーション用のオブジェクトを登録し、また、クライアントアプリケーション111からの入力情報をサーバアプリケーション101の入力形式に変換して転送し、また、サーバアプリケーション101からの出力情報をクライアントアプリケーション111に渡すべき出力形式に変換して転送する。

【0022】定義情報格納部141は、サーバアプリケーション101の入出力とクライアントアプリケーション111の入出力との間のデータ変換を定義する情報、汎用ラッパープログラム121と、サーバアプリケーション101およびクライアントアプリケーション111との間の通信に関する定義情報など、汎用ラッパープログラム121がラッピングする際に必要とする、サーバアプリケーション101およびクライアントアプリケーション111に固有な各種情報を格納する。オブジェクト管理サービス151は、分散オブジェクト環境上で使用できるオブジェクトの識別子や存在場所などの情報を管理し、分散オブジェクト環境内の各アプリケーションからの要求に応じて、分散オブジェクト環境で利用可能なオブジェクトの登録、削除、検索を行うことができる。分散オブジェクト環境内のアプリケーションが、オブジェクト管理サービス151にオブジェクトを登録することによって、その登録されたオブジェクトは分散オブジェクト環境内で利用可能な状態となり、また、分散オブジェクト環境内のアプリケーションが、利用したいオブジェクトの情報をオブジェクト管理サービス151に問い合わせ、情報を取得することで、そのアプリケーションは利用したいオブジェクトを呼び出すことができ

る。

【0023】図2、図3は、図1に示す本発明の一実施形態の動作を説明するために引用した図であり、それぞれ、制御手順をフローチャートで、動作概念をブロックで示した図である。以下、図2、図3を参照しながら図1に示す本発明一実施形態の動作について説明する。

【0024】まず、汎用ラッパープログラム121は、オブジェクト管理サービス151に、サーバアプリケーション用のオブジェクトを登録しておく(図2のステップA1)。クライアントアプリケーション101は、処理を要求するサーバアプリケーション101に対応するオブジェクト情報を、オブジェクト管理サービス151に問い合わせることによって、サーバアプリケーション用のオブジェクトの情報を取得する(ステップA2)。クライアントアプリケーション111は、取得した情報を用いて、サーバアプリケーション用のオブジェクトを呼び出すことで、汎用ラッパープログラム121にサーバアプリケーション101へ渡すためのデータが送信される(ステップA3)。

【0025】汎用ラッパープログラム121は、クライアントアプリケーション111からの入力データを受信し(ステップA4)、定義情報格納部141上の情報に基づいて、サーバアプリケーション101の入力形式にデータを変換する(ステップA5)。次に、汎用ラッパープログラム121は、変換した入力データを、サーバアプリケーション101のインターフェースに則った形式で送信し(ステップA6)、サーバアプリケーション101はこれを受信する(ステップA7)。入力データを受け取ったサーバアプリケーション101は、その入力データに基づいてプログラムを実行し(ステップA8)、実行結果を含む出力データを送信する(ステップA9)。

【0026】汎用ラッパープログラム121は、サーバアプリケーション101の出力データを受信し(ステップA10)、定義情報格納部141上の情報に基づいて、クライアントアプリケーション111が要求する出力データに変換する(ステップA11)。最後に、汎用ラッパープログラム121は変換した出力データを送信し(ステップA12)、クライアントアプリケーション111はこれを受信する(ステップA13)ことで、クライアントアプリケーション111からのサーバアプリケーション101の実行は完了する。

【0027】なお、上述した本発明実施形態において、汎用ラッパープログラム121そのものの動作は従来のラッパープログラム(図8、122)と変わらないが、サーバアプリケーション101の入出力とクライアントアプリケーション111が必要とする入出力とのデータ変換や、サーバアプリケーション101と通信するために必要な情報、オブジェクト管理サービス151に登録する情報など、クライアントアプリケーション111とサ

サーバアプリケーション101との間のやりとりに必要な情報に関して、汎用ラッパープログラム121自身が定義情報格納部141に格納されている情報を参照することになっているため、サーバアプリケーション101やクライアントアプリケーション111の仕様に変更が生じたときに、定義情報格納部141に格納されている情報を変更するだけで、汎用ラッパープログラム121の動作を変更することができる。

【0028】すなわち、汎用ラッパープログラム121自身には全く手を加えることなく、サーバアプリケーション101やクライアントアプリケーション111の変更を汎用ラッパープログラム121に反映することができる。また、上述した本発明実施形態では、さらに、サーバコンピュータ100、クライアントコンピュータ110、通信管理装置120、オブジェクト管理サーバコンピュータ150それぞれへの通信に関する定義情報を変更された場合も、汎用ラッパープログラム121自身には手を加えることなく通信可能にすることができる。それは、定義情報格納部141に、分散オブジェクト環境側および非分散オブジェクト環境側それぞれの通信に関する定義情報も含まれており、汎用ラッパープログラム121は、その情報を参照しながら通信手段を生成しているためである。

【0029】次に、具体例を用いて本発明実施形態の動作を詳細に説明する。図3に示すように、例えば、分散オブジェクト環境内の電文の形式が、電文1、12のように、上下方向とも、インターフェース名、オペレーション名、入出力の属性、型、値、属性、型、値、…という形式になっており、サーバアプリケーション用の電文が、上り電文は、電文2.1のように、関数名、入力値、入力値、…という形式で、下り電文が、電文2.2のように、関数名、出力値、出力値、…という形式であるとする。このとき、汎用ラッパープログラム121は、分散オブジェクト環境内上り電文1.1をサーバアプリケーション用上り電文2.1に、サーバアプリケーション用下り電文2.2を分散オブジェクト環境内下り電文1.2に変換する。電文1.1と電文2.2の間および電文2.1と電文2.2の間の変換規則を定義するのが、インターフェース定義情報31であり、定義情報格納部141に格納されている。そして、このインターフェース定義情報31が汎用ラッパープログラム121に反映される。

【0030】図3に示す例では、インターフェース定義情報31は、クライアント側メソッド名、クライアント側オペレーション名、クライアント側入力情報、クライアント側出力情報、サーバ側関数名、サーバ側入力情報、サーバ側出力情報の順に記述しており、クライアント側の情報とサーバ側の情報との間には“:”で区別しており、クライアント情報、サーバ情報ともに、識別名と、入力情報と出力情報との間には“:”で区別している。また、各情報に要素が複数ある場合は、その間

は“,”で区別している。

【0031】ここで、インターフェース定義情報31における定義例について、図3のインターフェース定義情報31の1行目を例に挙げて説明する。クライアントアプリケーション111で呼ぶ分散オブジェクトのインターフェース名がInterfaceであり、オペレーション名がMethodであった場合は、サーバアプリケーション101内のTransactionが呼び出されることを意味しており、また、クライアントからの2つの入力データの型とともにlong型であり、それをサーバアプリケーション101の引数にするときにはともにデータ型をINT4型に変換すること、また、サーバ側の出力データの型がINT4型であり、それをクライアントへの出力にする時には、データ型をlong型に変換することを意味している。

【0032】また、クライアントアプリケーション111と汎用ラッパープログラム121の間、および、汎用ラッパープログラム121とサーバアプリケーション101の間で通信を行うために必要な定義情報は、通信定義情報32として、定義情報格納部141に格納されている。図3に示すのは、通信定義情報32の1行目でオブジェクト管理サーバコンピュータのIPアドレスを、2行目でサーバアプリケーション101が存在するサーバコンピュータの識別名を定義している。インターフェース定義情報31および通信定義情報32を汎用ラッパープログラム121が参照することで、汎用ラッパープログラム121はオブジェクト管理サービス151へのオブジェクトの送受および、クライアントとサーバとの間やラッパーとサーバとの間の通信と電文の変換を行うことができる。

【0033】図4は本発明が採用されるラッピングシステムの他の実施形態(第2の実施形態)を示すブロック図である。図1に示す第1の実施形態との差異は、通信管理装置170が、図1に示す通信管理装置120の構成に加え、IDL(Interface Definition Language)生成手段171を有することである。IDL生成手段171は、定義情報格納部141に格納されているインターフェース定義情報を参照することで、分散オブジェクト環境上のインターフェースを定義するための情報であるIDLを生成する。なお、クライアントアプリケーション111を作成する場合は、IDLを翻訳することによって、クライアントアプリケーションのインターフェース部分に相当するソースコードを生成することができ、そのソースコードを基にして作成することができる。ここでは、分散オブジェクト環境上のインターフェース定義を自動的に作成することによって、クライアントアプリケーション111を作成する手間を軽減することができる。

【0034】なお、上述した図4に示す第2の実施形態では、通信管理装置170上でIDL生成手段171が

動作する場合について説明したが、I DL 生成手段171は、図1における定義情報格納部141内の情報を参照することができる任意のコンピュータ上で動作させてもよい。また、図1、図4に示すそれぞれ第1および第2の実施形態では、通信管理装置120上で汎用ラッパープログラム121が動作する場合について説明したが、図5に示すように、汎用ラッパープログラム121をサーバコンピュータ160上で動作させることもでき、また、図6に示すようにクライアントコンピュータ110上で動作させてもよい。

【0035】更に、図1、図4に示す第1および第2の実施形態では、分散オブジェクト環境側に存在するクライアントアプリケーションが非分散オブジェクト環境上に存在するサーバアプリケーションに対して処理を要求する形態について説明したが、非分散オブジェクト環境側に存在するアプリケーションが分散オブジェクト環境側に存在するアプリケーションに対して処理を要求する場合や、また、分散オブジェクト環境上に存在するアプリケーションと非分散オブジェクト環境上に存在するアプリケーションが相互に処理を要求する場合も同様である。また、第1および第2の実施形態では、オブジェクト管理サーバコンピュータ150上でオブジェクト管理サービス151が動作する場合について説明したが、オブジェクト管理サービス151は、図1におけるクライアントコンピュータ110や通信管理装置120や図5におけるサーバコンピュータ160を含めて分散オブジェクト環境上に存在する任意のコンピュータ上で動作させてもよい。

【0036】図7は、本発明が採用されるラッピングシステムの更に他の実施形態(第3の実施形態)を示すブロック図である。図7を参照すると、本発明における第3の実施形態は、汎用ラッパープログラムを記録した記録媒体400を備える。この記録媒体400は磁気ディスク、半導体メモリその他の記録媒体であってよい。汎用ラッパープログラムは記録媒体400から通信管理装置300に読み込まれ、通信管理装置300の動作を制御する。通信管理装置300は、汎用ラッパープログラムの制御により以下の処理、すなわち第1および第2の実施形態における通信管理装置120および170による処理と同一の処理、を実行する。

【0037】まず、汎用ラッパープログラムは、オブジェクト管理サービス151に、サーバアプリケーション用のオブジェクトを登録しておく。クライアントアプリケーション101は、処理を要求するサーバアプリケーション101に対応するオブジェクト情報を、オブジェクト管理サービス151に問い合わせることによって、サーバアプリケーション用のオブジェクトの情報を取得する。クライアントアプリケーション111は、取得した情報を用いてサーバアプリケーション用のオブジェクトを呼び出すことで、汎用ラッパープログラムにサーバアプリ

ケーション101へ渡すためのデータが送信される。次に、汎用ラッパープログラムは、クライアントアプリケーション111からの入力データを受信し、定義情報格納部141上の情報に基づいて、サーバアプリケーション101の入力形式にデータを変換する。そして、汎用ラッパープログラムは、変換した入力データを、サーバアプリケーション101のインターフェースに則った形式で送信し、サーバアプリケーション101はこれを受信する。

【0038】入力データを受け取ったサーバアプリケーション101は、その入力データに基づいてプログラムを実行し、実行結果を含む出力データを送信する。汎用ラッパープログラムは、サーバアプリケーション101の出力データを受信し、定義情報格納部141上の情報に基づいて、クライアントアプリケーション111が要求する出力データに変換する。最後に、汎用ラッパープログラムは変換した出力データを送信し、クライアントアプリケーション111はこれを受信することで、クライアントアプリケーション111からのサーバアプリケーション101の実行は完了する。また、記録媒体400に記憶されている汎用ラッパープログラムに、I DL 生成手段171が含まれている場合には、I DL 生成手段171は、定義情報格納部141に格納されているインターフェース定義情報を参照することで、分散オブジェクト環境上のインターフェースを定義するための情報であるI DL を生成する。なお、クライアントアプリケーション111を作成する場合は、このI DL の解釈手助けを用いることで、クライアントアプリケーションのインターフェース部分に相当するソースコードを生成することで、そのソースコードを基にして作成することができ

【0039】図1、図4および図7に示す、第1、第2および第3の実施形態では、処理の受け渡しを行うサーバアプリケーションとクライアントアプリケーションの対応が1対1の場合について説明したが、処理の受け渡しを行うサーバアプリケーションとクライアントアプリケーションの対応が1対多、多対1、多対多でもよく、その数に制限はない。また、この場合、サーバアプリケーションが動作するサーバコンピュータの数およびクライアントアプリケーションが動作するクライアントコンピュータの数も、それぞれ複数でもよく、その数に制限はない。

【0040】

【発明の効果】以上説明のように本発明によれば、ラッパープログラムがインターフェースの定義情報を参照して動作するため、その定義情報を変更するだけで、インターフェースの変更がラッパープログラムに反映され、したがって、非分散オブジェクト環境内のアプリケーションのインターフェースに変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、ラッパ

一プログラムに対してインターフェースの変更を反映できる。このため、非分散オブジェクト環境内のアプリケーションのインターフェースに変更が生じた際にプログラムの負担が大幅に軽減される。

【0041】また、同理由で分散オブジェクト環境内のアプリケーションのインターフェースに変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、ラッパープログラムに対してインターフェースの変更を反映できる。このため、分散オブジェクト環境内のアプリケーションのインターフェースに変更が生じた際にプログラムの負担が大幅に軽減される。

【0042】更に、ラッパープログラムが分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義情報を参照して動作するため、その定義情報を変更するだけで、通信定義の変更がラッパープログラムに反映されことから、分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義に変更が生じた際に、ラッパープログラム自身には何も手を加えることなく、ラッパープログラムに対して通信定義の変更を反映できる。このため、分散オブジェクト環境内および非分散オブジェクト環境内の通信に関する定義に変更が生じた際にプログラムの負担が大幅に軽減される。

【0043】また、定義情報格納部に格納されているインターフェース情報から、分散オブジェクト環境上でのインターフェース定義であるIDLを自動的に作成することによって、非分散オブジェクト環境内のアプリケーションを利用する分散オブジェクト環境内のクライアントアプリケーションを作成する手間を軽減することができるともできる。

【図面の簡単な説明】

【図1】 本発明における第1の実施形態の構成を示すブロック図である。

【図2】 図1に示す実施形態の動作を示すフローチャートである。

【図3】 図1に示す実施形態の動作の具体例を示す図である。

【図4】 本発明における第2の実施形態の構成を示すブロック図である。

【図5】 図1に示す第1の実施形態において、汎用ラッパープログラムがサーバコンピュータに含まれる場合の構成を示すブロック図である。

【図6】 図1に示す第1の実施形態において、汎用ラッパープログラムがクライアントコンピュータに含まれる場合の構成を示すブロック図である。

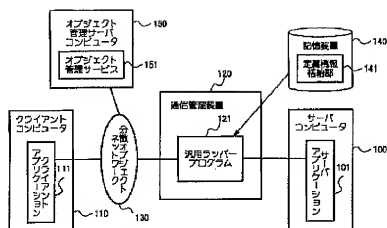
【図7】 本発明における第3の実施形態の構成を示すブロック図である。

【図8】 従来のラッピングシステムの構成例を示すブロック図である。

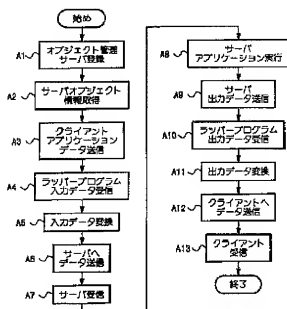
【符号の説明】

- 1 1 分散オブジェクト環境内上り電文
- 1 2 分散オブジェクト環境内下り電文
- 2 1 サーバアプリケーション用上り電文
- 2 2 サーバアプリケーション用下り電文
- 3 1 インターフェース定義情報
- 3 2 通信定義情報
- 100、160 サーバコンピュータ
- 101 サーバアプリケーション
- 110、180 クライアントコンピュータ
- 111 クライアントアプリケーション
- 120、170、300 通信管理装置
- 121 汎用ラッパープログラム
- 122 ラッパープログラム
- 123 ラッパーソースコード生成手段
- 124 ラッパーソースコード
- 125 コンパイラ
- 130 分散オブジェクトネットワーク
- 140 記憶装置
- 141 定義情報格納部
- 150 オブジェクト管理サーバコンピュータ
- 151 オブジェクト管理サーバ
- 171 IDL生成手段
- 400 記録媒体

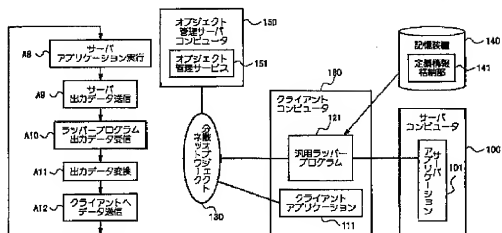
【 図1 】



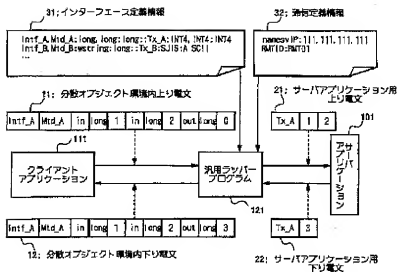
【 図2 】



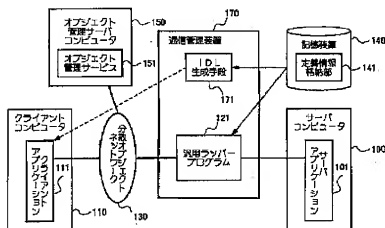
【 図6 】



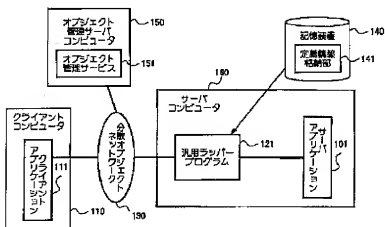
【 図3 】



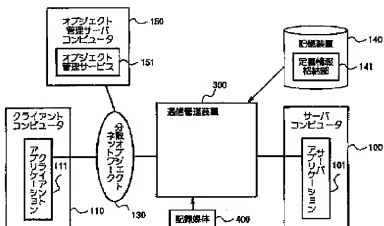
【 図4 】



【 図5 】



【 図7 】



【 図8 】

